

CONTENTS

1. INTRODUCTION

1.1 Overview

1.2 Purpose

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map

2.2 Ideation & Brainstorming Map

3. RESULT

4. ADVANTAGES & DISADVANTAGES

5. APPLICATIONS

6. CONCLUSION

7. FUTURE SCOPE

8. APPENDIX

8.1 Source Code

INTRODUCTION

The main objective of flexible email client apps, where you can customize and streamline your email experience like never before! Our email client app is designed with flexibility in mind, allowing you to tailor the app to meet your unique email needs.

1.1 Overview:

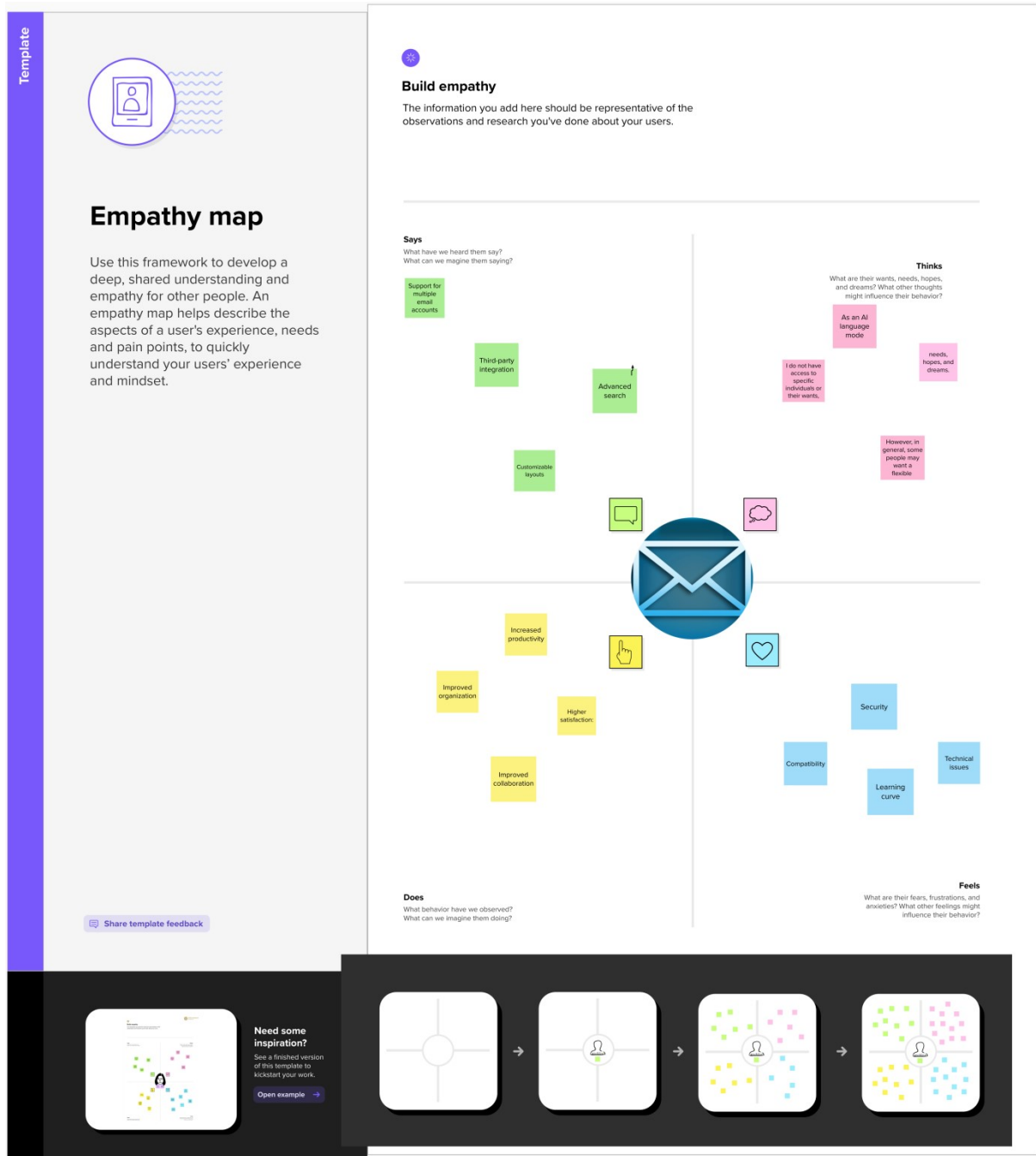
- Multiple account support: The app should allow users to add and manage multiple email accounts from different providers, such as Gmail, Yahoo, or Outlook.
- Customizable interface: Users should be able to customize the interface of the app to suit their preferences, including the ability to change the colour scheme, layout, and font size..
- Advanced search options: The app should provide advanced search options that allow users to find specific emails quickly and easily.
- Email filtering and sorting: Users should be able to filter and sort their emails based on different criteria, such as sender, subject, date, and attachments.

1.2 Purpose:

- Increased productivity: A flexible email client app can help users manage their emails more efficiently, allowing them to spend less time on email and more time on other important tasks.
- Customizability: A flexible email client app can be customized to suit the individual needs and preferences of users. This can include features such as customizable themes, font sizes, and email sorting options.
- Integration: A flexible email client app can integrate with other tools and apps, such as calendars, task managers, and project management software. This can help users streamline their workflows and stay organized.

PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map:



2.2 Ideation & Brainstorming Map:

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not doing it the same way.

- 0 minutes concept
- 10 minutes concept
- 15 minutes concept

Download template

Before you collaborate

A little bit of preparation goes a long way with brainstorming. Here are some things to do before you jump in.

- 1. **Get inspired** - Gather any inspiration from the internet and use it to fuel your ideas. (10 minutes)
- 2. **Write your problem statement** - Write down the problem you want to solve in a clear and concise way. (10 minutes)
- 3. **Brainstorm** - Write down any ideas that come to mind. (15 minutes)
- 4. **Group ideas** - Group your ideas into categories. (10 minutes)
- 5. **Prioritize** - Prioritize your ideas based on their potential. (10 minutes)
- 6. **After you collaborate** - Reflect on the session and share your ideas with the team. (10 minutes)

Define your problem statement

What problem are you trying to solve? Write it down in a clear and concise way. The goal is to define the problem you want to solve.

Example: **How can we make our university more sustainable?**

Brainstorm

Write down any ideas that come to mind. Use the table below to organize your ideas.

Problem Statement	Idea 1	Idea 2	Idea 3	Idea 4	Idea 5
How can we make our university more sustainable?	1. Use renewable energy	2. Reduce paper usage	3. Recycle waste	4. Use public transport	5. Plant trees
	6. Use energy-efficient light bulbs	7. Reduce water usage	8. Recycle food waste	9. Use carpooling	10. Use composting
	11. Use solar panels	12. Reduce plastic usage	13. Recycle electronic waste	14. Use bike-sharing	15. Use rainwater harvesting
	16. Use wind turbines	17. Reduce food waste	18. Recycle construction waste	19. Use car-sharing	20. Use greywater recycling
	21. Use geothermal energy	22. Reduce textile waste	23. Recycle paint and ink	24. Use ride-sharing	25. Use water-saving devices
	26. Use hydroelectric power	27. Reduce furniture waste	28. Recycle oil and grease	29. Use car-pooling	30. Use low-flow toilets
	31. Use biomass energy	32. Reduce electronics waste	33. Recycle tires	34. Use car-sharing	35. Use dual-flush toilets
	36. Use wind power	37. Reduce pharmaceutical waste	38. Recycle refrigerants	39. Use car-sharing	40. Use low-flow showerheads
	41. Use solar power	42. Reduce laboratory waste	43. Recycle fluorescent bulbs	44. Use car-sharing	45. Use water-saving faucets
	46. Use wind power	47. Reduce chemical waste	48. Recycle toner and ink	49. Use car-sharing	50. Use water-saving showerheads
	51. Use solar power	52. Reduce food waste	53. Recycle paint	54. Use car-sharing	55. Use water-saving showerheads
	56. Use wind power	57. Reduce electronic waste	58. Recycle motor oil	59. Use car-sharing	60. Use water-saving showerheads
	61. Use solar power	62. Reduce construction waste	63. Recycle antifreeze	64. Use car-sharing	65. Use water-saving showerheads
	66. Use wind power	67. Reduce medical waste	68. Recycle oil	69. Use car-sharing	70. Use water-saving showerheads
	71. Use solar power	72. Reduce hazardous waste	73. Recycle paint	74. Use car-sharing	75. Use water-saving showerheads
	76. Use wind power	77. Reduce radioactive waste	78. Recycle oil	79. Use car-sharing	80. Use water-saving showerheads
	81. Use solar power	82. Reduce nuclear waste	83. Recycle paint	84. Use car-sharing	85. Use water-saving showerheads
	86. Use wind power	87. Reduce chemical waste	88. Recycle oil	89. Use car-sharing	90. Use water-saving showerheads
	91. Use solar power	92. Reduce pharmaceutical waste	93. Recycle paint	94. Use car-sharing	95. Use water-saving showerheads
	96. Use wind power	97. Reduce laboratory waste	98. Recycle oil	99. Use car-sharing	100. Use water-saving showerheads
	99. Use solar power	100. Reduce medical waste	101. Recycle paint	102. Use car-sharing	103. Use water-saving showerheads
	100. Use wind power	101. Reduce hazardous waste	102. Recycle oil	103. Use car-sharing	104. Use water-saving showerheads
	101. Use solar power	102. Reduce radioactive waste	103. Recycle paint	104. Use car-sharing	105. Use water-saving showerheads
	102. Use wind power	103. Reduce chemical waste	104. Recycle oil	105. Use car-sharing	106. Use water-saving showerheads
	103. Use solar power	104. Reduce pharmaceutical waste	105. Recycle paint	106. Use car-sharing	107. Use water-saving showerheads
	104. Use wind power	105. Reduce laboratory waste	106. Recycle oil	107. Use car-sharing	108. Use water-saving showerheads
	105. Use solar power	106. Reduce medical waste	107. Recycle paint	108. Use car-sharing	109. Use water-saving showerheads
	106. Use wind power	107. Reduce hazardous waste	108. Recycle oil	109. Use car-sharing	110. Use water-saving showerheads
	107. Use solar power	108. Reduce radioactive waste	109. Recycle paint	110. Use car-sharing	111. Use water-saving showerheads
	108. Use wind power	109. Reduce chemical waste	110. Recycle oil	111. Use car-sharing	112. Use water-saving showerheads
	109. Use solar power	110. Reduce pharmaceutical waste	111. Recycle paint	112. Use car-sharing	113. Use water-saving showerheads
	110. Use wind power	111. Reduce laboratory waste	112. Recycle oil	113. Use car-sharing	114. Use water-saving showerheads
	111. Use solar power	112. Reduce medical waste	113. Recycle paint	114. Use car-sharing	115. Use water-saving showerheads
	112. Use wind power	113. Reduce hazardous waste	114. Recycle oil	115. Use car-sharing	116. Use water-saving showerheads
	113. Use solar power	114. Reduce radioactive waste	115. Recycle paint	116. Use car-sharing	117. Use water-saving showerheads
	114. Use wind power	115. Reduce chemical waste	116. Recycle oil	117. Use car-sharing	118. Use water-saving showerheads
	115. Use solar power	116. Reduce pharmaceutical waste	117. Recycle paint	118. Use car-sharing	119. Use water-saving showerheads
	116. Use wind power	117. Reduce laboratory waste	118. Recycle oil	119. Use car-sharing	120. Use water-saving showerheads
	117. Use solar power	118. Reduce medical waste	119. Recycle paint	120. Use car-sharing	121. Use water-saving showerheads
	118. Use wind power	119. Reduce hazardous waste	120. Recycle oil	121. Use car-sharing	122. Use water-saving showerheads
	119. Use solar power	120. Reduce radioactive waste	121. Recycle paint	122. Use car-sharing	123. Use water-saving showerheads
	120. Use wind power	121. Reduce chemical waste	122. Recycle oil	123. Use car-sharing	124. Use water-saving showerheads
	121. Use solar power	122. Reduce pharmaceutical waste	123. Recycle paint	124. Use car-sharing	125. Use water-saving showerheads
	122. Use wind power	123. Reduce laboratory waste	124. Recycle oil	125. Use car-sharing	126. Use water-saving showerheads
	123. Use solar power	124. Reduce medical waste	125. Recycle paint	126. Use car-sharing	127. Use water-saving showerheads
	124. Use wind power	125. Reduce hazardous waste	126. Recycle oil	127. Use car-sharing	128. Use water-saving showerheads
	125. Use solar power	126. Reduce radioactive waste	127. Recycle paint	128. Use car-sharing	129. Use water-saving showerheads
	126. Use wind power	127. Reduce chemical waste	128. Recycle oil	129. Use car-sharing	130. Use water-saving showerheads
	127. Use solar power	128. Reduce pharmaceutical waste	129. Recycle paint	130. Use car-sharing	131. Use water-saving showerheads
	128. Use wind power	129. Reduce laboratory waste	130. Recycle oil	131. Use car-sharing	132. Use water-saving showerheads
	129. Use solar power	130. Reduce medical waste	131. Recycle paint	132. Use car-sharing	133. Use water-saving showerheads
	130. Use wind power	131. Reduce hazardous waste	132. Recycle oil	133. Use car-sharing	134. Use water-saving showerheads
	131. Use solar power	132. Reduce radioactive waste	133. Recycle paint	134. Use car-sharing	135. Use water-saving showerheads
	132. Use wind power	133. Reduce chemical waste	134. Recycle oil	135. Use car-sharing	136. Use water-saving showerheads
	133. Use solar power	134. Reduce pharmaceutical waste	135. Recycle paint	136. Use car-sharing	137. Use water-saving showerheads
	134. Use wind power	135. Reduce laboratory waste	136. Recycle oil	137. Use car-sharing	138. Use water-saving showerheads
	135. Use solar power	136. Reduce medical waste	137. Recycle paint	138. Use car-sharing	139. Use water-saving showerheads
	136. Use wind power	137. Reduce hazardous waste	138. Recycle oil	139. Use car-sharing	140. Use water-saving showerheads
	137. Use solar power	138. Reduce radioactive waste	139. Recycle paint	140. Use car-sharing	141. Use water-saving showerheads
	138. Use wind power	139. Reduce chemical waste	140. Recycle oil	141. Use car-sharing	142. Use water-saving showerheads
	139. Use solar power	140. Reduce pharmaceutical waste	141. Recycle paint	142. Use car-sharing	143. Use water-saving showerheads
	140. Use wind power	141. Reduce laboratory waste	142. Recycle oil	143. Use car-sharing	144. Use water-saving showerheads
	141. Use solar power	142. Reduce medical waste	143. Recycle paint	144. Use car-sharing	145. Use water-saving showerheads
	142. Use wind power	143. Reduce hazardous waste	144. Recycle oil	145. Use car-sharing	146. Use water-saving showerheads
	143. Use solar power	144. Reduce radioactive waste	145. Recycle paint	146. Use car-sharing	147. Use water-saving showerheads
	144. Use wind power	145. Reduce chemical waste	146. Recycle oil	147. Use car-sharing	148. Use water-saving showerheads
	145. Use solar power	146. Reduce pharmaceutical waste	147. Recycle paint	148. Use car-sharing	149. Use water-saving showerheads
	146. Use wind power	147. Reduce laboratory waste	148. Recycle oil	149. Use car-sharing	150. Use water-saving showerheads
	147. Use solar power	148. Reduce medical waste	149. Recycle paint	150. Use car-sharing	151. Use water-saving showerheads
	148. Use wind power	149. Reduce hazardous waste	150. Recycle oil	151. Use car-sharing	152. Use water-saving showerheads
	149. Use solar power	150. Reduce radioactive waste	151. Recycle paint	152. Use car-sharing	153. Use water-saving showerheads
	150. Use wind power	151. Reduce chemical waste	152. Recycle oil	153. Use car-sharing	154. Use water-saving showerheads
	151. Use solar power	152. Reduce pharmaceutical waste	153. Recycle paint	154. Use car-sharing	155. Use water-saving showerheads
	152. Use wind power	153. Reduce laboratory waste	154. Recycle oil	155. Use car-sharing	156. Use water-saving showerheads
	153. Use solar power	154. Reduce medical waste	155. Recycle paint	156. Use car-sharing	157. Use water-saving showerheads
	154. Use wind power	155. Reduce hazardous waste	156. Recycle oil	157. Use car-sharing	158. Use water-saving showerheads
	155. Use solar power	156. Reduce radioactive waste	157. Recycle paint	158. Use car-sharing	159. Use water-saving showerheads
	156. Use wind power	157. Reduce chemical waste	158. Recycle oil	159. Use car-sharing	160. Use water-saving showerheads
	157. Use solar power	158. Reduce pharmaceutical waste	159. Recycle paint	160. Use car-sharing	161. Use water-saving showerheads
	158. Use wind power	159. Reduce laboratory waste	160. Recycle oil	161. Use car-sharing	162. Use water-saving showerheads
	159. Use solar power	160. Reduce medical waste	161. Recycle paint	162. Use car-sharing	163. Use water-saving showerheads
	160. Use wind power	161. Reduce hazardous waste	162. Recycle oil	163. Use car-sharing	164. Use water-saving showerheads
	161. Use solar power	162. Reduce radioactive waste	163. Recycle paint	164. Use car-sharing	165. Use water-saving showerheads
	162. Use wind power	163. Reduce chemical waste	164. Recycle oil	165. Use car-sharing	166. Use water-saving showerheads
	163. Use solar power	164. Reduce pharmaceutical waste	165. Recycle paint	166. Use car-sharing	167. Use water-saving showerheads
	164. Use wind power	165. Reduce laboratory waste	166. Recycle oil	167. Use car-sharing	168. Use water-saving showerheads
	165. Use solar power	166. Reduce medical waste	167. Recycle paint	168. Use car-sharing	169. Use water-saving showerheads
	166. Use wind power	167. Reduce hazardous waste	168. Recycle oil	169. Use car-sharing	170. Use water-saving showerheads
	167. Use solar power	168. Reduce radioactive waste	169. Recycle paint	170. Use car-sharing	171. Use water-saving showerheads
	168. Use wind power	169. Reduce chemical waste	170. Recycle oil	171. Use car-sharing	172. Use water-saving showerheads
	169. Use solar power	170. Reduce pharmaceutical waste	171. Recycle paint	172. Use car-sharing	173. Use water-saving showerheads
	170. Use wind power	171. Reduce laboratory waste	172. Recycle oil	173. Use car-sharing	174. Use water-saving showerheads
	171. Use solar power	172. Reduce medical waste	173. Recycle paint	174. Use car-sharing	175. Use water-saving showerheads
	172. Use wind power	173. Reduce hazardous waste	174. Recycle oil	175. Use car-sharing	176. Use water-saving showerheads
	173. Use solar power	174. Reduce radioactive waste	175. Recycle paint	176. Use car-sharing	177. Use water-saving showerheads
	174. Use wind power	175. Reduce chemical waste	176. Recycle oil	177. Use car-sharing	178. Use water-saving showerheads
	175. Use solar power	176. Reduce pharmaceutical waste	177. Recycle paint	178. Use car-sharing	179. Use water-saving showerheads
	176. Use wind power	177. Reduce laboratory waste	178. Recycle oil	179. Use car-sharing	180. Use water-saving showerheads
	177. Use solar power	178. Reduce medical waste	179. Recycle paint	180. Use car-sharing	181. Use water-saving showerheads
	178. Use wind power	179. Reduce hazardous waste	180. Recycle oil	181. Use car-sharing	182. Use water-saving showerheads
	179. Use solar power	180. Reduce radioactive waste	181. Recycle paint	182. Use car-sharing	183. Use water-saving showerheads
	180. Use wind power	181. Reduce chemical waste	182. Recycle oil	183. Use car-sharing	184. Use water-saving showerheads
	181. Use solar power	182. Reduce pharmaceutical waste	183. Recycle paint	184. Use car-sharing	185. Use water-saving showerheads
	182. Use wind power	183. Reduce laboratory waste	184. Recycle oil	185. Use car-sharing	186. Use water-saving showerheads
	183. Use solar power	184. Reduce medical waste	185. Recycle paint	186. Use car-sharing	187. Use water-saving showerheads
	184. Use wind power	185. Reduce hazardous waste	186. Recycle oil	187. Use car-sharing	188. Use water-saving showerheads
	185. Use solar power	186. Reduce radioactive waste	187. Recycle paint	188. Use car-sharing	189. Use water-saving showerheads
	186. Use wind power	187. Reduce chemical waste	188. Recycle oil	189. Use car-sharing	190. Use water-saving showerheads
	187. Use solar power	188. Reduce pharmaceutical waste	189. Recycle paint	190. Use car-sharing	191. Use water-saving showerheads
	188. Use wind power	189. Reduce laboratory waste	190. Recycle oil	191. Use car-sharing	192. Use water-saving showerheads
	189. Use solar power	190. Reduce medical waste	191. Recycle paint	192. Use car-sharing	193. Use water-saving showerheads
	190. Use wind power	191. Reduce hazardous waste	192. Recycle oil	193. Use car-sharing	194. Use water-saving showerheads
	191. Use solar power	192. Reduce radioactive waste	193. Recycle paint	194. Use car-sharing	195. Use water-saving showerheads
	192. Use wind power	193. Reduce chemical waste	194. Recycle oil	195. Use car-sharing	196. Use water-saving showerheads
	193. Use solar power	194. Reduce pharmaceutical waste	195. Recycle paint	196. Use car-sharing	197. Use water-saving showerheads
	194. Use wind power	195. Reduce laboratory waste	196. Recycle oil	197. Use car-sharing	198. Use water-saving showerheads
	195. Use solar power	196. Reduce medical waste	197. Recycle paint	198. Use car-sharing	199. Use water-saving showerheads
	196. Use wind power	197. Reduce hazardous waste	198. Recycle oil	199. Use car-sharing	200. Use water-saving showerheads
	197. Use solar power	198. Reduce radioactive waste	199. Recycle paint	200. Use car-sharing	201. Use water-saving showerheads
	198. Use wind power	199. Reduce chemical waste	200. Recycle oil	201. Use car-sharing	202. Use water-saving showerheads
	199. Use solar power	200. Reduce pharmaceutical waste	201. Recycle paint	202. Use car-sharing	203. Use water-saving showerheads
	200. Use wind power	201. Reduce laboratory waste	202. Recycle oil	203. Use car-sharing	204. Use water-saving showerheads
	201. Use solar power	202. Reduce medical waste	203. Recycle paint	204. Use car-sharing	205. Use water-saving showerheads
	202. Use wind power	203. Reduce hazardous waste	204. Recycle oil	205. Use car-sharing	206. Use water-saving showerheads
	203. Use solar power	204. Reduce radioactive waste	205. Recycle paint	206. Use car-sharing	207. Use water-saving showerheads
	204. Use wind power	205. Reduce chemical waste	206. Recycle oil	207. Use car-sharing	208. Use water-saving showerheads
	205. Use solar power	206. Reduce pharmaceutical waste	207. Recycle paint	208. Use car-sharing	209. Use water-saving showerheads
	206. Use wind power	207. Reduce laboratory waste	208. Recycle oil	209. Use car-sharing	210. Use water-saving showerheads
	207. Use solar power	208. Reduce medical waste	209. Recycle paint	210. Use car-sharing	211. Use water-saving showerheads
	208. Use wind power	209. Reduce hazardous waste	210. Recycle oil	211. Use car-sharing	212. Use water-saving showerheads
	209. Use solar power	210. Reduce radioactive waste	211. Recycle paint	212. Use car-sharing	213. Use water-saving showerheads
	210. Use wind power	211. Reduce chemical waste	212. Recycle oil	213. Use car-sharing	214. Use water-saving showerheads
	211. Use solar power	212. Reduce pharmaceutical waste	213. Recycle paint	214. Use car-sharing	215. Use water-saving showerheads
	212. Use wind power	213. Reduce laboratory waste	214. Recycle oil	215. Use car-sharing	216. Use water-saving showerheads
	213. Use solar power	214. Reduce medical waste	215. Recycle paint	216. Use car-sharing	217. Use water-saving showerheads
	214. Use wind power	215. Reduce hazardous waste	216. Recycle oil	217. Use car-sharing	218. Use water-saving showerheads
	215. Use solar power	216. Reduce radioactive waste	217. Recycle paint	218. Use car-sharing	219. Use water-saving showerheads
	216. Use wind power	217. Reduce chemical waste	218. Recycle oil	219. Use car-sharing	220. Use water-saving showerheads
	217. Use solar power	218. Reduce pharmaceutical waste	219. Recycle paint	220. Use car-sharing	221. Use water-saving showerheads
	218. Use wind power	219. Reduce laboratory waste	220. Recycle oil	221. Use car-sharing	222. Use water-saving showerheads
	219. Use solar power	220. Reduce medical waste	221. Recycle paint	222. Use car-sharing	223. Use water-saving showerheads
	220. Use wind power	221. Reduce hazardous waste	222. Recycle oil	223. Use car-sharing	224. Use water-saving showerheads
	221. Use solar power	222. Reduce radioactive waste	223. Recycle paint	224. Use car-sharing	225. Use water-saving showerheads
	222. Use wind power	223. Reduce chemical waste	224. Recycle oil	225. Use car-sharing	226. Use water-saving showerheads
	223. Use solar power	224. Reduce pharmaceutical waste	225. Recycle paint	226. Use car-sharing	227. Use water-saving showerheads
	224. Use wind power	225. Reduce laboratory waste	226. Recycle oil	227. Use car-sharing	228. Use water-saving showerheads
	225. Use solar power	226. Reduce medical waste	227. Recycle paint	228. Use car-sharing	229. Use water-saving showerheads
	226. Use wind power	227. Reduce hazardous waste	228. Recycle oil	229. Use car-sharing	230. Use water-saving showerheads
	227. Use solar power	228. Reduce radioactive waste	229. Recycle paint	230. Use car-sharing	231. Use water-saving showerheads
	228. Use wind power	229. Reduce chemical waste	230. Recycle oil	231. Use car-sharing	232. Use

log in Page:

RESULT



Login

Username

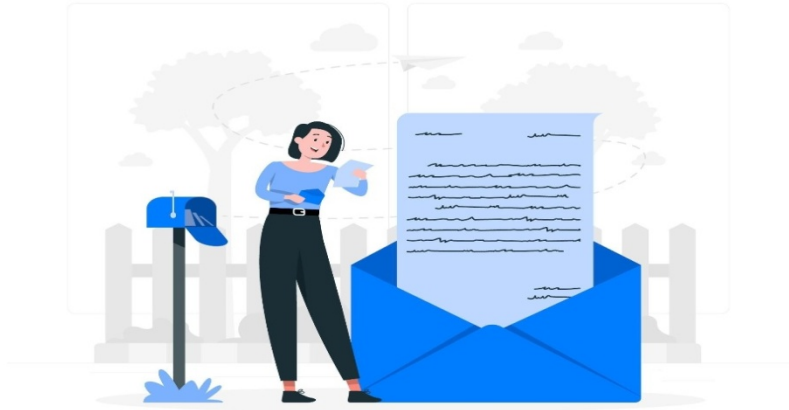
Password

Login

Sign up

Forget password?

Register page



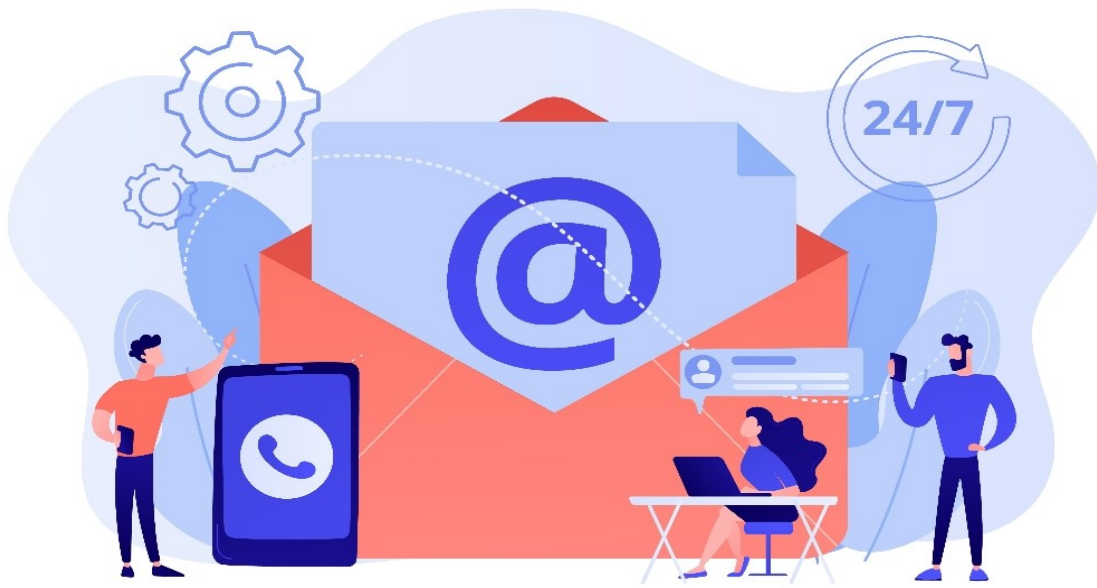
Register

Register

Have an account? [Log in](#)

Home Page:

Home Screen



Send Email

View Emails

Send mail page

Send Mail

Receiver Email-Id

Email address

Mail Subject

Subject

Mail Body

Body

Send Email

ADVANTAGES & DISADVANTAGES

1.1 Advantages:

- **Integration:** A flexible email client app can integrate with other tools and apps, such as calendars, task managers, and project management software. This can help users streamline their workflows and stay organized..
- **Productivity:** A flexible email client app can help users manage their emails more efficiently, allowing them to spend less time on email and more time on other important tasks. This can include features such as automated replies, email scheduling, and snooze options.
- **Accessibility:** A flexible email client app can be designed to be accessible to all users, including those with disabilities. This can include features such as text-to-speech and keyboard shortcuts.

1.2 Disadvantages:

- **Compatibility issues:** A flexible email client app may not be compatible with all email providers or email formats. This can limit its usefulness for some users and require them to switch to a different app.
- **Cost:** Some flexible email client apps may require users to pay for certain customization options or features, which can add up over time and make the app less accessible for some users.
- **Overall,** while a flexible email client app can offer many benefits, it is important for users to carefully consider their needs and the potential drawbacks before committing to using one.

APPLICATIONS

- **Apple Mail:** The default email client app on Apple devices, which offers advanced filtering options, the ability to customize swipe gestures, and integration with other Apple tools and services such as iCloud.
- **Microsoft Outlook:** A popular email client app that integrates with other Microsoft tools and services, such as Office 365, OneDrive, and Skype. It offers advanced filtering options, calendar integration, and the ability to create custom rules for managing emails.
- **Gmail:** Google's email client app, which offers advanced filtering options, labels, and integration with other Google tools and services. It also offers a powerful search function and the ability to snooze emails for later.
- **Thunderbird:** A free and open-source email client app that offers a high level of customization options, including themes, add-ons, and the ability to customize keyboard shortcuts. It also supports multiple email accounts and protocols.
- **Spike:** A modern email client app that combines email, messaging, and collaboration features into a single platform. It offers advanced search options, the ability to create groups and share files, and an AI-powered smart inbox.

Conclusion

We conclude that the flexible email client app is an essential tool for anyone who needs to manage their email accounts on the go. By providing users with a range of features and options that can be customized to meet their specific needs and preferences, a flexible email client app can help users stay organized, save time, and communicate more effectively with their contacts. Whether you're a busy professional, a student, or anyone else who relies on email for communication, a flexible email client app can be an invaluable asset.

Future Scopes

In future adding As the world becomes more conscious of environmental issues, email client apps will need to become more sustainable in their design and operation. This could include features such as energy-efficient modes, reduced data usage, and the ability to integrate with eco-friendly email hosting services. Overall, the future of flexible email client applications will be shaped by a combination of new technologies, changing user needs and preferences, and broader societal trends.

APPENDIX

Source Code : <https://github.com/MuthuSahin/Emailapp>

Code : -

1.LoginActivity.kt

```
package com.example.emailapplication
```

```
import android.content.Context
```

```
import android.content.Intent
```

```
import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.Image
```

```
import androidx.compose.foundation.background
```

```
import androidx.compose.foundation.layout.*
```

```
import androidx.compose.material.*
```

import androidx.compose.runtime.*

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.layout.ContentScale

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.text.font.FontFamily

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.input.PasswordVisualTransformation

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.core.content.ContextCompat

import com.example.emailapplication.ui.theme.EmailApplicationTheme

```
class LoginActivity : ComponentActivity() {  
  
    private lateinit var databaseHelper: UserDatabaseHelper  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
  
        super.onCreate(savedInstanceState)  
  
        databaseHelper = UserDatabaseHelper(this)  
  
        setContent {  
  
            LoginScreen(this, databaseHelper)  
  
        }  
  
    }  
  
    }  
  
    @Composable  
  
    fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {
```



```
var username by remember { mutableStateOf("") }
```

```
var password by remember { mutableStateOf("") }
```

```
var error by remember { mutableStateOf("") }
```

```
Column(
```

```
    modifier = Modifier.fillMaxSize().background(Color.White),
```

```
    horizontalAlignment = Alignment.CenterHorizontally,
```

```
    verticalArrangement = Arrangement.Center
```

```
) {
```

```
    Image(
```

```
        painterResource(id = R.drawable.email_login), contentDescription =
```

```
        ""
```

```
    )
```

```
Text(  
  
    fontSize = 36.sp,  
  
    fontWeight = FontWeight.ExtraBold,  
  
    fontFamily = FontFamily.Cursive,  
  
    text = "Login"  
  
)  
  
Spacer(modifier = Modifier.height(10.dp))
```

```
TextField(  
  
    value = username,  
  
    onChange = { username = it },  
  
    label = { Text("Username") },  
  
    modifier = Modifier.padding(10.dp)  
  
        .width(280.dp)  
  
)
```

```
TextField(  
  
    value = password,  
  
    onValueChange = { password = it },  
  
    label = { Text("Password") },  
  
    visualTransformation = PasswordVisualTransformation(),  
  
    modifier = Modifier.padding(10.dp)  
  
    .width(280.dp)  
  
)
```

```
if (error.isNotEmpty()) {  
  
    Text(  
  
        text = error,  
  
        color = MaterialTheme.colors.error,  
  
        modifier = Modifier.padding(vertical = 16.dp)  
  
    )
```

```
)  
  
}
```

```
Button(
```

```
onClick = {
```

```
if (username.isNotEmpty() && password.isNotEmpty()) {
```

```
val user = databaseHelper.getUserByUsername(username)
```

```
if (user != null && user.password == password) {
```

```
error = "Successfully log in"
```

```
context.startActivity(
```

```
Intent(
```

```
context,
```

```
MainActivity::class.java
```

```
)
```

```
)
```

```

        //onLoginSuccess()

    }

    } else {

        error = "Please fill all fields"

    }

},

    colors = ButtonDefaults.buttonColors(backgroundColor =
Color(0xFFd3e5ef)),

    modifier = Modifier.padding(top = 16.dp)

) {

    Text(text = "Login")

}

Row {

    TextButton(onClick = {context.startActivity(

```

```
Intent(  
  
    context,  
  
    RegisterActivity::class.java  
  
    )  
  
})  
  
)  
  
{ Text(color = Color(0xFF31539a),text = "Sign up") }  
  
TextButton(onClick = {  
  
    })  
  
  
    {  
  
        Spacer(modifier = Modifier.width(60.dp))  
  
        Text(color = Color(0xFF31539a),text = "Forget password?")  
  
    }  
  
}
```

```
}
```

```
}
```

```
private fun startMainPage(context: Context) {
```

```
    val intent = Intent(context, MainActivity::class.java)
```

```
    ContextCompat.startActivity(context, intent, null)
```

```
}
```

2.MainActivity.kt

```
package com.example.emailapplication
```

```
import android.content.Context
```

```
import android.content.Intent
```

```
import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.Image
```

```
import androidx.compose.foundation.background
```

```
import androidx.compose.foundation.layout.*
```

```
import androidx.compose.material.*
```

```
import androidx.compose.runtime.Composable
```

```
import androidx.compose.ui.Alignment
```

```
import androidx.compose.ui.Modifier
```


import androidx.compose.ui.graphics.Color

import androidx.compose.ui.layout.ContentScale

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.core.content.ContextCompat

import androidx.core.content.ContextCompat.startActivity

import com.example.emailapplication.ui.theme.EmailApplicationTheme

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

```
// A surface container using the 'background' color from the theme
```

```
Surface(
```

```
    modifier = Modifier.fillMaxSize().background(Color.White),
```

```
) {
```

```
    Email(this)
```

```
}
```

```
}
```

```
}
```

```
}
```

```
@Composable
```

```
fun Email(context: Context) {
```

```
    Text(
```

```
        text = "Home Screen",
```

```
        modifier = Modifier.padding(top = 74.dp, start = 100.dp, bottom = 24.dp),

        color = Color.Black,

        fontWeight = FontWeight.Bold,

        fontSize = 32.sp

    )

    Column(

        horizontalAlignment = Alignment.CenterHorizontally,

        verticalArrangement = Arrangement.Center

    ) {

        Image(

            painterResource(id = R.drawable.home_screen), contentDescription =

            ""

        )
```

```
Button(onClick = {  
  
    context.startActivity(  
  
        Intent(  
  
            context,  
  
            SendMailActivity::class.java  
  
        )  
  
    )  
  
    },  
  
    colors = ButtonDefaults.buttonColors(backgroundColor =  
Color(0xFFadbef4))  
  
    ) {  
  
        Text(  
  
            text = "Send Email",  
  
            modifier = Modifier.padding(10.dp),
```

```
        color = Color.Black,

        fontSize = 15.sp

    )

}

Spacer(modifier = Modifier.height(20.dp))

Button(onClick = {

    context.startActivity(

        Intent(

            context,

            ViewMailActivity::class.java

        )

    )

},
```

```
        colors = ButtonDefaults.buttonColors(backgroundColor =
Color(0xFFadbef4))

    ) {

        Text(

            text = "View Emails",

            modifier = Modifier.padding(10.dp),

            color = Color.Black,

            fontSize = 15.sp

        )

    }

}

}
```

3.RegisterActivity.kt

```
package com.example.emailapplication
```

```
import android.content.Context
```

```
import android.content.Intent
```

```
import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.Image
```

```
import androidx.compose.foundation.background
```

```
import androidx.compose.foundation.layout.*
```

```
import androidx.compose.material.*
```

```
import androidx.compose.runtime.*
```

```
import androidx.compose.ui.Alignment
```

```
import androidx.compose.ui.Modifier
```

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.layout.ContentScale

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.text.font.FontFamily

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.input.PasswordVisualTransformation

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import androidx.core.content.ContextCompat

import com.example.emailapplication.ui.theme.EmailApplicationTheme

class RegisterActivity : ComponentActivity() {

private lateinit var databaseHelper: UserDatabaseHelper

override fun onCreate(savedInstanceState: Bundle?) {


```
super.onCreate(savedInstanceState)
```

```
databaseHelper = UserDatabaseHelper(this)
```

```
setContent {
```

```
RegistrationScreen(this, databaseHelper)
```

```
}
```

```
}
```

```
}
```

```
@Composable
```

```
fun RegistrationScreen(context: Context, databaseHelper:
```

```
UserDatabaseHelper) {
```

```
var username by remember { mutableStateOf("") }
```

```
var password by remember { mutableStateOf("") }
```

```
var email by remember { mutableStateOf("") }
```

```
var error by remember { mutableStateOf("") }
```

```
Column(
```

```
    modifier = Modifier.fillMaxSize().background(Color.White),
```

```
    horizontalAlignment = Alignment.CenterHorizontally,
```

```
    verticalArrangement = Arrangement.Center
```

```
) {
```

```
    Image(
```

```
        painterResource(id = R.drawable.email_signup), contentDescription =
```

```
        "",
```

```
        modifier = Modifier.height(300.dp)
```

```
)
```

```
Text(  
  
    fontSize = 36.sp,  
  
    fontWeight = FontWeight.ExtraBold,  
  
    fontFamily = FontFamily.Cursive,  
  
    text = "Register"  
  
)
```

```
Spacer(modifier = Modifier.height(10.dp))
```

```
TextField(  
  
    value = username,  
  
    onValueChange = { username = it },  
  
    label = { Text("Username") },  
  
    modifier = Modifier  
  
        .padding(10.dp)  
  
        .width(280.dp)
```

)

TextField(

value = email,

onValueChange = { email = it },

label = { Text("Email") },

modifier = Modifier

.padding(10.dp)

.width(280.dp)

)

TextField(

value = password,

onValueChange = { password = it },

```
label = { Text("Password") },

visualTransformation = PasswordVisualTransformation(),

modifier = Modifier

    .padding(10.dp)

    .width(280.dp)

)

if (error.isNotEmpty()) {

    Text(

        text = error,

        color = MaterialTheme.colors.error,

        modifier = Modifier.padding(vertical = 16.dp)

    )

}
```

```
Button(  
  
    onClick = {  
  
        if (username.isNotEmpty() && password.isNotEmpty() &&  
email.isNotEmpty()) {  
  
            val user = User(  
  
                id = null,  
  
                firstName = username,  
  
                lastName = null,  
  
                email = email,  
  
                password = password  
  
            )  
  
            databaseHelper.insertUser(user)  
  
            error = "User registered successfully"  
  
            // Start LoginActivity using the current context
```

```
context.startActivity(  
  
    Intent(  
  
        context,  
  
        LoginActivity::class.java  
  
    )  
  
    )  
  
    )  
  
    } else {  
  
        error = "Please fill all fields"  
  
    }  
  
    },  
  
    colors = ButtonDefaults.buttonColors(backgroundColor =  
    Color(0xFFd3e5ef),  
  
    modifier = Modifier.padding(top = 16.dp)  
  
    ) {
```

```
Text(text = "Register")
```

```
}
```

```
Spacer(modifier = Modifier.width(10.dp))
```

```
Spacer(modifier = Modifier.height(10.dp))
```

```
Row() {
```

```
Text(
```

```
    modifier = Modifier.padding(top = 14.dp), text = "Have an  
account?"
```

```
)
```

```
TextButton(onClick = {
```

```
    context.startActivity(
```

```
        Intent(
```

```
            context,
```

```
            LoginActivity::class.java
```



```
        )

    )

})

{

    Spacer(modifier = Modifier.width(10.dp))

    Text(color = Color(0xFF31539a),text = "Log in")

}

}

}

}

private fun startLoginActivity(context: Context) {

    val intent = Intent(context, LoginActivity::class.java)

    ContextCompat.startActivity(context, intent, null)

}
```

4.SendMainActivity.kt

```
package com.example.emailapplication
```

```
import android.annotation.SuppressLint
```

```
import android.content.Context
```

```
import android.content.Intent
```

```
import android.os.Bundle
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.layout.*
```

```
import androidx.compose.material.*
```

```
import androidx.compose.runtime.*
```

```
import androidx.compose.ui.Alignment
```

```
import androidx.compose.ui.Modifier
```

```
import androidx.compose.ui.graphics.Color
```

```
import androidx.compose.ui.platform.LocalContext
```

```
import androidx.compose.ui.text.TextStyle
```

```
import androidx.compose.ui.text.font.FontWeight
```

```
import androidx.compose.ui.text.style.TextAlign
```

```
import androidx.compose.ui.tooling.preview.Preview
```

```
import androidx.compose.ui.unit.dp
```

```
import androidx.compose.ui.unit.sp
```

```
import com.example.emailapplication.ui.theme.EmailApplicationTheme
```

```
class SendMailActivity : ComponentActivity() {
```

```
    private lateinit var databaseHelper: EmailDatabaseHelper
```

```
    @SuppressWarnings("UnusedMaterialScaffoldPaddingParameter")
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
```

```
        super.onCreate(savedInstanceState)
```

```
        databaseHelper = EmailDatabaseHelper(this)
```

```
setContent {  
  
    Scaffold(  
  
        // in scaffold we are specifying top bar.  
  
        topBar = {  
  
            // inside top bar we are specifying  
  
            // background color.  
  
            TopAppBar(backgroundColor = Color(0xFFadbef4), modifier =  
Modifier.height(80.dp),  
  
            // along with that we are specifying  
  
            // title for our top bar.  
  
            title = {  
  
                // in the top bar we are specifying  
  
                // title as a text  
  
                Text(  

```

// on below line we are specifying

// text to display in top app bar.

text = "Send Mail",

fontSize = 32.sp,

color = Color.Black,

// on below line we are specifying

// modifier to fill max width.

modifier = Modifier.fillMaxWidth(),

// on below line we are

// specifying text alignment.

textAlign = TextAlign.Center,

)

}

```
        )

    }

    ){

        // on below line we are

        // calling method to display UI.

        openEmailer(this,databaseHelper)

    }

}

}

}

}

}

@Composable

fun openEmailer(context: Context, databaseHelper: EmailDatabaseHelper) {

    // in the below line, we are

    // creating variables for URL
```

```
var recevierMail by remember {mutableStateOf("") }
```

```
var subject by remember {mutableStateOf("") }
```

```
var body by remember {mutableStateOf("") }
```

```
var error by remember { mutableStateOf("") }
```

```
// on below line we are creating
```

```
// a variable for a context
```

```
val ctx = LocalContext.current
```

```
// on below line we are creating a column
```

```
Column(
```

```
    // on below line we are specifying modifier
```

```
    // and setting max height and max width
```

```
    // for our column
```

```
    modifier = Modifier
```

.fillMaxSize()

.padding(top = 55.dp, bottom = 25.dp, start = 25.dp, end = 25.dp),

horizontalAlignment = Alignment.Start $) \{$

// on the below line, we are

```
// creating a text field.
```

```
Text(text = "Receiver Email-Id",
```

```
fontWeight = FontWeight.Bold,
```

fontSize = 16.sp)

TextField(

// on below line we are specifying

```
// value for our text field.
```

value = recevierMail,

// on below line we are adding on value

// change for text field.

onValueChange = { recevierMail = it },

// on below line we are adding place holder as text

label = { Text(text = "Email address") },

placeholder = { Text(text = "abc@gmail.com") },

// on below line we are adding modifier to it

// and adding padding to it and filling max width

modifier = Modifier

.padding(16.dp)

.fillMaxWidth(),

// on below line we are adding text style

// specifying color and font size to it.

textStyle = TextStyle(color = Color.Black, fontSize = 15.sp),

// on below line we are

// adding single line to it.

singleLine = true,

)

// on below line adding a spacer.

Spacer(modifier = Modifier.height(10.dp))

Text(text = "Mail Subject",

fontWeight = FontWeight.Bold,

fontSize = 16.sp)

// on the below line, we are creating a text field.

TextField(

// on below line we are specifying

// value for our text field.

value = subject,

// on below line we are adding on value change

// for text field.

onValueChange = { subject = it },

// on below line we are adding place holder as text

placeholder = { Text(text = "Subject") },

// on below line we are adding modifier to it

// and adding padding to it and filling max width

modifier = Modifier

.padding(16.dp)

```
.fillMaxWidth(),
```

```
// on below line we are adding text style
```

```
// specifying color and font size to it.
```

```
textStyle = TextStyle(color = Color.Black, fontSize = 15.sp),
```

```
// on below line we are
```

```
// adding single line to it.
```

```
singleLine = true,
```

```
)
```

```
// on below line adding a spacer.
```

```
Spacer(modifier = Modifier.height(10.dp))
```

```
Text(text = "Mail Body",
```

fontWeight = FontWeight.Bold,

fontSize = 16.sp)

// on the below line, we are creating a text field.

TextField(

// on below line we are specifying

// value for our text field.

value = body,

// on below line we are adding on value

// change for text field.

onValueChange = { body = it },

// on below line we are adding place holder as text

placeholder = { Text(text = "Body") },

// on below line we are adding modifier to it

// and adding padding to it and filling max width

modifier = Modifier

.padding(16.dp)

.fillMaxWidth(),

// on below line we are adding text style

// specifying color and font size to it.

textStyle = TextStyle(color = Color.Black, fontSize = 15.sp),

// on below line we are

// adding single line to it.

singleLine = true,

)

// on below line adding a spacer.

Spacer(modifier = Modifier.height(20.dp))

// on below line adding a

// button to send an email

Button(onClick = {

**if(recevierMail.isNotEmpty() && subject.isNotEmpty() &&
body.isNotEmpty()) {**

val email = Email(

id = null,

recevierMail = recevierMail,

subject = subject,

body = body

```
)

    databaseHelper.insertEmail(email)

    error = "Mail Saved"

} else {

    error = "Please fill all fields"

}


// on below line we are creating

// an intent to send an email

val i = Intent(Intent.ACTION_SEND)


// on below line we are passing email address,

// email subject and email body

val emailAddress = arrayOf(recevierMail)

i.putExtra(Intent.EXTRA_EMAIL,emailAddress)
```



```
i.putExtra(Intent.EXTRA_SUBJECT,subject)
```

```
i.putExtra(Intent.EXTRA_TEXT,body)
```

```
// on below line we are
```

```
// setting type of intent
```

```
i.setType("message/rfc822")
```

```
// on the below line we are starting our activity to open email  
application.
```

```
ctx.startActivity(Intent.createChooser(i,"Choose an Email client : "))
```

```
},
```

```
colors = ButtonDefaults.buttonColors(backgroundColor =  
Color(0xFFd3e5ef))
```

```
) {
```

// on the below line creating a text for our button.

Text(

// on below line adding a text ,

// padding, color and font size.

text = "Send Email",

modifier = Modifier.padding(10.dp),

color = Color.Black,

fontSize = 15.sp

)

}

}

5.ViewMailActivity.kt

```
package com.example.emailapplication
```

```
import android.annotation.SuppressLint
```

```
import android.os.Bundle
```

```
import android.util.Log
```

```
import androidx.activity.ComponentActivity
```

```
import androidx.activity.compose.setContent
```

```
import androidx.compose.foundation.Image
```

```
import androidx.compose.foundation.layout.*
```

```
import androidx.compose.foundation.layout.R
```

```
import androidx.compose.foundation.lazy.LazyColumn
```

```
import androidx.compose.foundation.lazy.LazyRow
```

```
import androidx.compose.foundation.lazy.items
```

```
import androidx.compose.material.*
```

import androidx.compose.runtime.Composable

import androidx.compose.ui.Modifier

import androidx.compose.ui.graphics.Color

import androidx.compose.ui.layout.ContentScale

import androidx.compose.ui.res.painterResource

import androidx.compose.ui.text.font.FontWeight

import androidx.compose.ui.text.style.TextAlign

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.unit.dp

import androidx.compose.ui.unit.sp

import com.example.emailapplication.ui.theme.EmailApplicationTheme

class ViewMailActivity : ComponentActivity() {

private lateinit var emailDatabaseHelper: EmailDatabaseHelper

@SuppressLint("UnusedMaterialScaffoldPaddingParameter")

```
override fun onCreate(savedInstanceState: Bundle?) {

    super.onCreate(savedInstanceState)

    emailDatabaseHelper = EmailDatabaseHelper(this)

    setContent {

        Scaffold(

            // in scaffold we are specifying top bar.

            topBar = {

                // inside top bar we are specifying

                // background color.

                TopAppBar(backgroundColor = Color(0xFFadbef4), modifier =

Modifier.height(80.dp),

                // along with that we are specifying

                // title for our top bar.

                title = {
```

// in the top bar we are specifying

// title as a text

Text(

// on below line we are specifying

// text to display in top app bar.

text = "View Mails",

fontSize = 32.sp,

color = Color.Black,

// on below line we are specifying

// modifier to fill max width.

modifier = Modifier.fillMaxWidth(),

// on below line we are

// specifying text alignment.

```
        textAlign = TextAlign.Center,

    )

    }

)

}

){

    val data = emailDatabaseHelper.getAllEmails();

    Log.d("swathi", data.toString())

    val email = emailDatabaseHelper.getAllEmails()

    ListListScopeSample(email)

}

}

}

}

@Composable
```

```
fun ListListScopeSample(email: List<Email>) {  
  
    LazyRow(  
  
        modifier = Modifier  
  
            .fillMaxSize(),  
  
        horizontalArrangement = Arrangement.SpaceBetween  
  
    ) {  
  
        item {  
  
            LazyColumn {  
  
                items(email) { email ->  
  
                    Column(  
  
                        modifier = Modifier.padding(  
  
                            top = 16.dp,  
  
                            start = 48.dp,  
  
                            bottom = 20.dp
```


)

) {

Text("Receiver_Mail: \${email.recevierMail}", fontWeight =
FontWeight.Bold)

Text("Subject: \${email.subject}")

Text("Body: \${email.body}")

}

}

}

}

}

}